

####【置顶】相关资料与网站汇集  
[ChainBook/IPFS-For-Chinese](#)  
[IPFS指南-知乎专栏](#)

####【置顶】文档  
[Kademlia协议原理简介.pdf](#)

####【置顶】TODO:

=====  
=====  
####【置顶】[IPFS-词汇表](#)  
####【置顶】[IPFS-常用调试命令](#)  
####【置顶】[ChainBook/IPFS-For-Chinese/IPFS Commands](#)(ipfs命令行简介)  
####【置顶】[chinanf-boy/go-ipfs-zh\(go-ipfs中文翻译\)](#)

####ipfs读源码  
1.参数解析  
比如终端上执行ipfs daemon  
root.go init() 准备了Root.Subcommands/RootRO.Subcommands 包含大部分command  
ipfs.go init() localCommands 准备了command: daemon/init/commands;  
main.go  
    main  
        mainRet  
            setupInterruptHandler 信号接收和处理、填充context  
            cli.Run (run.go)  
                Parse()->parse() (parse.go) 填充Request的Command、Arguments、Options等成员  
                    makeExecutor (main.go) 准备cmdDetails(默认值初始化为ipfs.go中)、http.client(如果需要的话), 返回cmds.Executor  
                    exctr.Execute (executor.go) 执行具体cmd的preRun/PostRun/Run  
2.daemon启动流程  
daemonFunc (daemon.go) 准备core.BuildCfg、  
    fsrepo.Open 准备文件锁 ~/ipfs/repo.lock  
    core.NewNode (daemon.go) 准备IpfsNode、  
        setupNode (builder.go) 准备BlockStore/gcBlockstore/  
            startOnlineServices (core.go)  
                startOnlineServicesWithHost 准备BitSwapNetwork/  
                routingOption 启动routing service  
                bitswap.New (bitswap.go)  
                go bs.wm.Run (wangtmanager.go)  
                startWorkers (workers.go)  
                    providerQueryManager (e := <-bs.findKeys)  
                    taskWorker  
                    rebroadcastWorker (bs.findKeys <- &blockRequest)  
                    provideCollector  
                    provideWorker  
                bserv.New (builder.go) 准备BlockService  
                dag.NewDAGService 准备dagService  
                pin.LoadPinner  
                resolver.NewBasicResolver  
                startLateOnlineServices  
    printSwarmAddrs (daemon.go)  
    serveHTTPApi  
    mountFuse 如果有需要的话  
    serveHTTPGateway  
3.get file流程

####有关NAT  
[P2P技术简介 - 崇山峻岭 - 博客园](#)  
检测当前网络NAT类型: <https://github.com/trilety/pystun>  
[检测nat类型流程图](#)  
[NAT原理与NAT穿透 - 深之JohnChen的专栏 - CSDN博客](#)  
[P2P技术详解\(-\): NAT详解——详细原理、P2P简介-网络编程/专项技术区 - 即时通讯开发者社区!](#)  
[UDP打洞 \(成功打洞\) - HotIce0 - CSDN博客](#)

####bitswap  
[Bitswap原理及实现](#)

####客户端模式DHT路由(--routing=dhtclient)、及其他实验阶段特性  
<https://github.com/ipfs/go-ipfs/blob/master/docs/experimental-features.md>  
<https://github.com/chinanf-boy/go-ipfs-zh/blob/master/docs/experimental-features.zh.md>

####storj白皮书-英文版  
[storj-whitepaper-v3](#)  
from: <https://github.com/storj/whitepaper>

####有关白皮书的部分解析  
[Understanding the IPFS White Paper part 1](#)

####  
[IPFS世界的物流系统: libp2p - IPFS Newb的博客 - CSDN博客](#)

####  
[go-libp2p 入门之 PingService example](#)

####libp2p原理及核心组件  
[比肩IPFS和Filecoin, 为什么说libp2p将成为区块链和物联网的基础设施?](#)

####有关数据存储细节  
[IPFS内部原理入门 \(An Introduction to IPFS\) - yichigo的专栏 - CSDN博客](#)  
[IPFS的存储与读取 - 区块链](#)

####集群搭建  
[IPFS系列 集群搭建 三](#)

####有关Kademlia协议  
[Kademlia协议原理简介.pdf](#)

####有关内存精简的初期调研  
1.Limiting the amount of memory go-ipfs may use (<https://discuss.ipfs.io/t/limiting-the-amount-of-memory-go-ipfs-may-use/3211/3>)  
1.1 ipfs本身使用只有200-300M, 但由于碎片化操作系统无法回收

1.2使用Swarm.ConnMgr.LowWater、Swarm.ConnMgr.HighWater限制节点维持连接数, 从而节省内存。目前在mac、android电视上LowWater/HighWater 默认都是600/900。  
参见config说明文档: <https://github.com/ipfs/go-ipfs/blob/master/docs/config.md#connmgr>

1.3 提到了systemd的开源软件, 用于限制最大使用内存。大致看了下, systemd是要取代initd, 称为linux启动的第一个进程的, 使用成本会很高。

1.4 没太懂的两个功能禁用, 还不知道影响如何。

1.5 ulimit 限制进程使用系统资源的工具, 超限后进程被终止。

2.Measuring memory usage #5530 (<https://github.com/ipfs/go-ipfs/issues/5530>)  
ps显示使用256, pprof显示使用了 26M的问题  
My OS is telling me that I'm using around 756MB:

```
$ ps aux | head -n1
USER          PID    %CPU %MEM    VSZ   RSS  TT  STAT STARTED      TIME COMMAND
$ ps aux | grep ipfs
rob           16719    0.5  3.1 558590836 256180 s002  Ss+   2:24pm   6:28.99 ipfs daemon
$
```

But pprof tells me that it's only using 76.38MB:

```
$ curl localhost:5001/debug/pprof/heap > ipfs.heap
% Total    % Received  Xferd  Average Speed   Time    Time     Time  Current
           % of total                               --:--:-- --:--:-- --:--:--
100 78010    0 78010    0      0 3822k    0 --:--:-- --:--:-- --:--:-- 4009k
$ go tool pprof /Users/rob/golang/bin/ipfs ipfs.heap
File: ipfs_space
Type: inuse_space
Time: Sep 27, 2018 at 5:21pm (AEST)
Entering interactive mode (type "help" for commands, "o" for options)
(pprof) top5
Showing nodes accounting for 38.29MB, 50.14% of 76.38MB total
Showing top 5 nodes out of 207
      flat  fat   sum%       cum   cum%
  13.47MB  17.63%  17.63%    22.97MB  30.07%  gx/ipfs/Qm0JMHF8ptRgx4E57UFH1T4YM6kqaJeYxZIMCDX
   9.50MB  12.44%  30.07%    9.50MB  12.44%  container/list.(*List).PushFront
   6.31MB   8.26%  38.34%    6.31MB   8.26%  gx/ipfs/Qmda4CPRvSRyox3SggJN6Df5ZGU5THuffPTp9uX
   4.52MB   5.91%  44.25%    4.52MB   5.91%  gx/ipfs/QmZt87ZHYGaZFBrTGPEqjJAC2yNAhg1RD8mu8KA
   4.50MB   5.89%  50.14%    4.50MB   5.89%  gx/ipfs/QmYnsdtJ3HsodkePE3eU3TsCs2YvPZJ4LoXnKi
(pprof)
```

2.1内存增长的原因, 碎片化, gc结束前分配内存

3.手机app中使用ipfs方案: IPFS in mobile use case #68  
(<https://github.com/ipfs/notes/issues/68>)

在源码中搜lowpower找到相关配置内容及使用说明:

```
147 > "lowpower": {
148 >   "Description": "Reduces daemon overhead on the system. May affect node
149 functionality - performance of content discovery and data
150 fetching may be degraded.
151 >
152 >   Transform: func(c "Config") error {
153 >     c.Routing.Type = "dhtclient"
154 >     c.Reprovider.Interval = "0"
155 >
156 >     c.Swarm.ConnMgr.LowWater = "0"
157 >     c.Swarm.ConnMgr.HighWater = 40
158 >     c.Swarm.ConnMgr.GracePeriod = time.Minute.String()
159 >     return nil
160 >   },
161 > },
162 > },
NORMAL > SPELL > +0 -0 -0 master > 1:profile.go
```

跟pc上默认配置相比, 不仅降低连接节点数, 延长免疫期限, type也有basic变成了dhtclient。

```
125 "Swarm": {
126   "AddrFilters": null,
127   "ConnMgr": {
128     "GracePeriod": "20s",
129     "HighWater": 900,
130     "LowWater": 600,
131     "Type": "basic"
132   },
133   "DisableBandwidthMetrics": false,
134   "DisableNatPortMap": false,
135   "DisableRelay": false,
136   "EnableRelayHop": false
137 }
138 }
NORMAL > SPELL > 1:config
~/ipfs/config 138L, 5296C
```

使用说明:  
26 To better support low-power devices, we've added a low-power config profile.  
27 This can be enabled when initializing a repo by running 'ipfs init' with the  
28 '--profile=lowpower' flag or later by running 'ipfs config profile apply lowpower'.  
29

4.Low Power Mode #4137 (<https://github.com/ipfs/go-ipfs/issues/4137>)

lowpower中的配置中包含了这两项配置内容。

5.Sweet-IPFS(<https://discuss.ipfs.io/t/sweet-ipfs-full-ipfs-node-for-android/3779/16>)  
app代码: <https://github.com/RHazDev/Sweet-IPFS/tree/master/app>  
一个文件分享软件, 使用的是kotlin语言写的, 非常简捷的几个文件, 可以看到使用的就是上面提到的lowpower模式的配置。

在电视上安装了apk, 很多功能都未实现。

```
Thread{
    val exec = run("init")
    Thread {
        exec.inputStream.bufferedReader().forEachLine { println(it) }
    }.start()
    Thread {
        exec.errorStream.bufferedReader().forEachLine { println(it) }
    }.start()
    exec.waitFor()
}
config{
    it.getAsJsonObject("Swarm").getAsJsonObject("ConnMgr").apply {
        remove("LowWater")
        addProperty("LowWater", 20)
        remove("HighWater")
        addProperty("HighWater", 40)
        remove("GracePeriod")
        addProperty("GracePeriod", "120s")
    }
}
progress.dismiss()
act.runOnUiThread(callback)
}.start()
```

6.textile-go(<https://github.com/textileio/>)  
一个图片分享软件, 提供android app使用的aar库文件, 代码: <https://github.com/textileio/textile-mobile>

7.IPFS light clients(<https://github.com/ipfs/notes/issues/206>)  
iphonex上能耗问题  
使用lowpower模式有所好转但仍然很高

提到有dht相关问题fix, 以及bitswap优化改善不少。(<https://github.com/ipfs/go-ipfs/issues/3786>)

8.The minimum requirements for the server are configured(<https://github.com/ipfs/go-ipfs/issues/4145>)  
解释了--routing=dhtclient及ipfs config Reprovider.Strategy pinned 会有什么影响。  
还有就是ipfs/config中swarm参数配置使用, 反馈是正确配置后内存使用情况好很多, 但仍然在缓慢增长。

####windows平台编译/运行  
[go-ipfs/docs/windows.md](#)

####有关mac、电视上环境搭建  
[004 IPFS - ABPS - 乐视Wiki](#)

参考:  
[\[IPFS + 区块链 系列\] 入门篇 - IPFS环境配置及博客搭建](#)

[打包Golang并在Android环境下运行 - CSDN博客](#)